UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/885,448	06/19/2001	Kenneth J. Hines	10488/11:1	2736

7590 07/09/2007

MICHAEL R. BARRE
BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025

EXAMINER	
WANG, JUE S	

ART UNIT	PAPER NUMBER
2193	

MAIL DATE	DELIVERY MODE
07/09/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	09/885,448	HINES, KENNETH J.	
	Examiner	Art Unit	
	Jue S. Wang	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *12 April 2007*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1,3-5,18,20,31,33-36 and 38* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1,3-5,18,20,31,33-36 and 38* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *19 February 2002* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1, 3-5, 18, 20, 31, 33-36, and 38 have been examined.

Claim Objections

2. Claims 18, 20, 36, and 38 are objected to because the claimed invention is directed to non-statutory subject matter.

3. Claims 18 and 36 are objected to because the claimed invention is directed to non-statutory subject matter. In claims 18 and 36, a "software system design tool" is recited; however, it appears that the "software system design tool" would reasonably be interpreted by one of ordinary skill in the art as software, per se, since the simulator, template tool, debugging tool recited as part of the design tool would reasonably be interpreted by one of ordinary skill in the art as software, per se. As such, it is believed that the "software system design tool" of claims 18 and 36 are reasonably interpreted as functional descriptive material, per se, failing to be tangibly embodied or include any recited hardware as part of the design tool. Claims 20 and 38 recite additional features of the "software system design tool" of claims 18 and 36 and they do not invalidate the reasonable interpretation of the "software system design tool" as software, per se.

Appropriate corrections are required.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 1, 3-5, 18, 20, 31, 33-36, and 38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Malin et al. (US 4,965,743, hereinafter Malin) in view of Bates, "Debugging Heterogeneous Distributed Systems Using Event-Based Models of Behavior", further in view of Brodsky et al. (US 5,960,199, hereinafter Brodsky).

6. As per claim 1, Malin teaches the invention as claimed including a method comprising:

generating a record of software system events, each event record with the record of system events representing an inter-component control or dataflow interaction ("The results of the simulation can either be permanently recorded in a log file or debug text.. ." in column 13 lines 34-36)

creating a behavioral template based on a predetermined behavior of the software system ("the library designer is able to create the knowledge representation information that is needed for the creation of models and the simulation of such models" in column 15 lines 44-47), wherein the predetermined behavior comprises a predetermined set of state changes selected from an execution of the software system, wherein the predetermined set of state changes represent coherent units of behavior by the system software ("Discrete event modeling and simulation is

characterized by state changes in a system's entities, 'events', that occur.. ." in column 4, lines 14-1 6. Further, note Figure 15 and the corresponding sections of the disclosure. "the method Run [model] in which the discrete event simulator runs the model by executing events on the event queue until the queue is empty" in column 25 lines 17-19. The events are predetermined state changes, as there is a predetermination concerning placing events in the queue, which further represent coherent units of behavior by the software system, as an event is indicative of some action or behavior by the system.)

identifying an occurrence of the predetermined behavior within the record of software system events, based on the behavioral template ("Additional analysis is obtained by comparison of log files to specify the differences in outcomes.. ." in col. 10 lines 52-53), wherein the predetermined set of state changes may be used as a behavioral model for a debugger to recognize (The state changes are represented as a behavioral model as noted above, and further, the user of the system of Malin has access to a debug facility as noted in column 29 lines 8-9, which gives debug information for the system.)

Malin does not teach replacing a found instance of a predetermined behavior with a replacement sequence of events, wherein the replacement sequence of events is an abstract even of a higher level than one or more system events that comprise the predetermined behavior.

Bates teaches an analogous behavior based modeling and debugging system the ability recognize stream of system events, and abstract the recognized behaviors into a high-level abstract event ("When a user-defined behavior model is successfully matched to the event stream, this derived behavior is abstracted into a representative high-level event..." on page 5,

section 2.1. Since the behavior is abstracted ***into*** (emphasis added) a representative high-level event, the high-level event replaces the behavior.)

It would have been obvious to one of ordinary skill in the art at the time the invention to consolidate a series of events in the system disclosed by Malin according to the abstraction techniques as taught by Bates, as this would enable a user of Malin's system to manage complexity, help organize the search for errors, and enhance the operation of debugging tools for distributed system (see page 2, section 1 of Bates). Furthermore, abstraction would formalize the modeling process and help to focus a tool user's attention on significant behaviors (see the paragraph bridging pages 2 and 3 of Bates).

Malin further teaches creating the behavioral template comprises creating a visual prototype (see column 24, lines 58-63). Malin and Bates do not teach that the visual prototype is created from an execution trace.

Brodsky teaches creating a visual prototype from an execution trace (see abstract, lines 1-7, 9-10, Figs 2A-3, column 3, lines 55-60, and column 4, lines 20-39; EN: the graphical trace view generated from an execution trace of the model provides a visual prototype).

It would have been obvious to one of ordinary skill in the art at the time the invention to modify Malin and Bates to create a visual prototype from an execution trace as taught by Brodsky because trace viewing allows programmers to validate and debug the execution of an object model and graphical representations of the traces are desirable since textual traces of many functions can be confusing (see column 2, lines 16-27 and column3, lines 55-60 of Brodsky).

7. As per claim 3, Malin further teaches creating the behavioral template comprises creating a behavior expression, which represents the predetermined behavior of the software system as claimed ("Statements are associated with processes ...They are written in terms of the operators, component variables inherited from the VCs, and the values of the valueclasses defined in the language" in column 24 lines 36-30).

8. As per claim 4, Malin further teaches simulating an execution of the software system, with the record of software system events generated by the simulator as claimed ("The results of the simulation can either be permanently recorded in a log file or debug text.. ." in column 13 lines 34-36).

9. As per claim 5, Malin further teaches that generating the record of software system events comprises:

instrumenting the software system to provide an event notification to a runtime operating system for each software system event, deploying the software system to a target architecture; and on the target architecture, capturing all notifications from the software system and storing the event notifications to create a record of software system events as claimed (Note Figure 1, item 132.1. To perform a trace of the system, and for the Log file to have been created, the system inherently had instrumentation to provide event notification so that the events could be recorded. Further, the simulation is inherently running on a target architecture.)

10. As per claim 18, Malin teaches the invention as claimed, including a software system design tool comprising:

a simulator for simulating an execution of the software system (Note Figure 1, item 13 and the corresponding section of the disclosure)

a template tool for creating a behavioral template based on a predetermined behavior of the software system ("the library designer is able to create the knowledge representation information that is needed for the creation of models and the simulation of such models" in column 15, lines 44-47), wherein the predetermined behavior comprises a predetermined set of state changes selected from an execution of the software system, wherein the predetermined set of state changes represent coherent units of behavior by the system software ("Discrete event modeling and simulation is characterized by state changes in a system's entities, 'events', that occur.. ." in column 4 lines 14-16. Further, note Figure 15 and the corresponding sections of the disclosure. "the method Run [model] in which the discrete event simulator runs the model by executing events on the event queue until the queue is empty" in col. 25 lines 17-19. The events are predetermined state changes, as there 1s a predetermination concerning placing events in the queue, which further represent coherent units of behavior by the software system, as an event is indicative of some action or behavior by the system.) and wherein the predetermined set of state changes may be used as a behavioral model for a debugger to recognize (The state changes are represented as a behavioral model as noted above, and further, the user of the system of Malin has access to a debug facility as noted in column 29 lines 8-9, which gives debug information for the system.)

a debugging tool for identifying an instance of the predetermined behavior of the software system from a simulated execution of the software system based on the behavioral template ("Additional analysis is obtained by comparison of log files to specify the differences in outcomes ..." in col. 10 h c s 52-53. Further, "the debug facility allows the user to turn debug on or off ..." in col. 29 lines 8-9)

Malin does not teach replacing a found instance of a predetermined behavior with a replacement sequence of events, wherein the replacement sequence of events is an abstract even of a higher level than one or more system events that comprise the predetermined behavior.

Bates teaches an analogous behavior based modeling and debugging system the ability recognize a stream of system events, and abstract the recognized behaviors into a high-level abstract event ("When a user-defined behavior model is successfully matched to the event stream, this derived behavior is abstracted into a representative high-level event..." on page 5, section 2.1. Since the behavior is abstracted ***into*** (emphasis added) a representative high-level event, the high-level event replaces the behavior.)

It would have been obvious to one of ordinary skill in the art at the time the invention to consolidate a series of events in the system disclosed by Malin according to the abstraction techniques as taught by Bates, as this would enable a user of Malin's system to manage complexity, help organize the search for errors, and enhance the operation of debugging tools for distributed system (see page 2, section 1 of Bates). Furthermore, abstraction would formalize the modeling process and help to focus a tool user's attention on significant behaviors (see the paragraph bridging pages 2 and 3 of Bates).

Malin further teaches that the template tool allows creating the behavioral template based on a visual prototype (see column 24, lines 58-63). Malin and Bates do not teach that the visual prototype is from an execution trace.

Brodsky teaches creating a visual prototype from an execution trace (see abstract, lines 1-7, 9-10, Figs 2A-3, column 3, lines 55-60, and column 4, lines 20-39; EN: the graphical trace view generated from an execution trace of the model provides a visual prototype).

It would have been obvious to one of ordinary skill in the art at the time the invention to modify Malin and Bates to create a behavioral template based on a visual prototype from an execution trace as taught by Brodsky because trace viewing allows programmers to validate and debug the execution of an object model and graphical representations of the traces are desirable since textual traces of many functions can be confusing (see column 2, lines 16-27 and column3, lines 55-60 of Brodsky).

11. As per claim 20, Malin further teaches that the template tool allows a designer to create a behavioral template based on a behavior expression ("Statements are associated with processes.. .They are written in terms of the operators, component variables inherited from the VCs, and the values of the valueclasses defined in the language" in column 24 lines 36-30).

12. As per claims 31, 33-36, and 38, the limitations recited in these claims are substantially similar to those recited in claims 1, 3-5, 18, and 20. Therefore, they are rejected using the same reasons as claims 1, 3-5, 18, and 20.

Response to Arguments

13. Applicant's arguments with respect to claims 1, 3-5, 18, 20, 31, 33-36, and 38 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

14. **THIS ACTION IS MADE FINAL**. Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Jue S. Wang whose telephone number is (571) 270-1655. The examiner can normally be reached on M-Th 7:30 am - 5:00pm (EST).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Meng-Ai An can be reached on 571-272-3756. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

J.W.
7/2/2007

MENG-AL T. AN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100